UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,694	01/02/2004	Robert DeSantis	IBMP014/SVL920040508US1	3817

63056 7590 06/01/2007

MOLLBORN PATENTS
ATTN: IBM
2840 COLBY DRIVE
BOULDER, CO 80305

EXAMINER	
NGUYEN, PHILLIP H	
ART UNIT	**PAPER NUMBER**
2191	

MAIL DATE	DELIVERY MODE
06/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/750,694	DESANTIS, ROBERT	
	Examiner	Art Unit	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 March 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to amendment filed on March 26, 2007.

2. As per Applicant's request, claims 1-5 and 8-20 have been amended, claims 1-20 remain pending and have been considered below.

Specification

3. The amendment filed on 3/26/2007 overcomes the objection set forth to the specification of previous action. Therefore, the objection is withdrawn.

Claim Objections

4. The amendment filed on 3/26/2007 overcomes the claim objections set forth to claims 1-20 of previous action. Therefore, claim objections are withdrawn.

Claim Rejections - 35 USC § 112

5. The amendment filed on 3/26/2007 is not deemed persuasive to overcome the rejection set forth to claims 1-4 and 11-20 for containing trademark/trade name JavaScript™/Java™. Therefore, Examiner maintains the rejection.

- Trademark or Trade Name as a Limitation in the Claims: Claims 1-4 and 11-20 contain the trademarks or trade names Java/JavaScript™. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See Ex parte Simpson, 218 USPQ 1020 (Bd.

App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the good associated with the trademark or trade name. In the present case, the trademark or trade name is used to identify or describe a family of products generated in the propriety programming language call JavaScript™ and accordingly, the identification or description is indefinite.

Response to Arguments

6. Applicant's arguments filed 3/26/2007 have been fully considered but they are not deemed persuasive.

Applicant asserts on page 6 of the amendment that trademark or trade name is allowed in the claims if the product to which the trademark refers is set forth in such language that its identity is clear.

Examiner respectfully disagrees, as cited under 35 USC 112, above, the claim scope is uncertain because trademark or trade name cannot be used properly to identify any particular material or product (Java or JavaScript). Applicant is suggested to delete Java or JavaScript from the claims. For example, claim 1 recites, "...JAVASCRIPT™ interpreted script language..." should be changed to "...interpreted script language..." Examiner is authorized to use trademark in the specification (not in the claims) as long as the proprietary nature of the marks should be respected and every effort made to

prevent their use in any manner, which might adversely affect their validity as trademarks.

Applicant asserts on page 8 of the amendment that the API in Nelson's approach does not invoked by clients, but instead the clients communicate with a web server in a conventional way, which in turn invokes an interface module 23, which in turn invokes the cube interface object.

Examiner respectfully disagrees with all the allegations as argued. Examiner in his previous office action gave detailed explanation of claimed limitations and pointed out some example locations in the cited prior art for assisting Applicant to locate the relevant limitations disclosed by the prior art. The APIs in Nelson's approach were created on the web server 13, which supported by cube interface object 24 (see at least FIG. 4). This cube interface object 24 interacts with user/client in response to user request (see at least col. 9, line 4-10 "**In response to a user request to expand a member, web browser 33 sends an expand request to cube interface object 24. In one embodiment web browser 33 accesses an Active Server Page corresponding to an expand request, which invokes a corresponding method of cube interface object 24**..."). This indicates that APIs are invoked by user/client. In the case the APIs are not invoked directly from the user/client, they are indirectly invoked by user/client thru interface module 23.

Applicant further asserts on page 8 of the amendment that Nelson does not teach "automatically creating a JavaScript interpreted script language program that

contains calls to the server-side software method in accordance with the Application Program Interface."

Examiner respectfully disagrees with the allegation as argued. Examiner in his previous office action gave detailed explanation of claimed limitations and pointed out few locations in the cited prior for assisting Applicant to locate the relevant limitations disclosed by the prior art. Nelson discloses, "...**translates report object 15 into a client side script that, when executed by the client, builds a client side representation of report object 15.**" The script (JavaScript) is created from the report object 15 on the server and transmitted to client for generating a report. Cube interface object 24 is manipulating and controlling the report object 15. Therefore, this report object 15 contains calls to cube interface object 24 in order for the cube interface object 24 to manipulate and control the report object 15.

Applicant asserts on page 9 of the amendment that Merrill is described in term of a Visual Basic Script not JavaScript.

Examiner respectfully disagrees with all the allegations as argued. Merrill discloses "**This includes applications written in a scripting language (e.g., text files processed at run-time) or executable files compiled from conventional languages such as C, C++, or Java**" col. 35, line 60-63, and also see col. 36, line 4-5. Examiner only needs to show why it would have been obvious to execute the script program on the client side to call the server side software method.

Applicant further asserts on page 9 that Examiner needs to show the motivation for combining Nelson and Merrill's approaches. It would have been obvious to one

having an ordinary skill in the art to recognize that JavaScript is client side language that runs on the client browser. The server in Merrill's approach is acting as a central repository for holding data/methods (routines). Instead of each client keeps a copy of the methods (this is redundancy), the server keeps the original methods and the clients can access the methods by calling them. This prevents from having code redundant stored at multiple places.

Applicant asserts on page 10 that Mikhail does not teach, "identifying the registered server-side methods".

Examiner respectfully disagrees with the allegation as argued. The process of registering itself with the server is also considered as identifying itself to the server.

Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See MPEP 2111 [R-1] Interpretation of Claims-Broadest Reasonable Interpretation. During patent examining, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.'

Applicant always has the opportunity to amend the claims during the prosecution and broad interpretation by the examiner reduce the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541, 550-51 (CCPA 1969).

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1 and 4 are rejected under 35 U.S.C. 102(b) as being anticipated by Nelson (Patent No.: US 7,093,194 B2).

As per claim 1:

Nelson discloses:

- although, Nelson does not explicitly disclose creating an API on the server side for a server-side software method. It is inherent in Nelson ("**API and TABLE I**" Col 5, line 31-65);

- automatically creating a JavaScript™ program that contains calls to the server-side software method in accordance with the API ("**converter 27 generates the client-side script**" Col 8, line 53-54); and

- sending the created JavaScript™ program to the client side ("**transmits the script to web browser via network**" Col 8, line 54-55).

As per claim 4:

Nelson discloses:

- wherein the JavaScript™ is executed by a non-modified standard browser program ("**Upon receiving the script, client device 9 executes the script and builds a presentation model 35" Col 8, line 55-57, this indicates that the client executes the script to build a presentation without modified itself**).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2, 12-13, and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Merrill et al. (Patent No.: US 6,369,821 B2).

As per claim 2:

Nelson does not explicitly disclose:

- executing the script on the client side to call the server side software method.

However, Merrill discloses:

- executing the script on the client side to call the server side software method (see at least col. 35, line 15 "**to execute the script code, the browser used the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code...when the script code references the character control, the browser accesses the animation server**" - **which means, when executing the script on the client side, it invokes the server so that the browser can access the server**).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include the invoking the server process of Merrill in Nelson's approach. One of the ordinary skilled in the art would have been motivated to include the invoking the server process in Nelson because the browser can access the animation server's methods and properties of a particular character to the animation server (Col 35, line 15-16).

As per claim 12:

Merrill disclose:

- wherein executing the JavaScript™ includes converting the parameters sent to the server side ("**Clients invoke this method to instruct the server to generate speech output for a specified text string. Clients specify a text**

string, which the speech output engine converts into digitized audio output. The animation server plays clip..." Col 23, line 1-5).

As per claim 13:

Merrill discloses:

- wherein executing the JavaScript™ includes converting results sent from the server side ("**The server generates this event when it encounters a bookmark tag in a text string as it converts the text string into speech output. The client can insert this tag in the text string provided with a Speak method**" Col 24, line 58-61).

As per claim 17:

Nelson discloses:

- receiving a JavaScript™ program that is automatically created by the server-side ("**converter 27 generates the client-side script, packet engine 21 transmits the script to web browser 33**" Col 8, line 53-55).
- executing the JavaScript™ on the client side ("**upon receiving the script, client device 9 executes the script**" Col 8, line 55-56).

Nelson does not explicitly disclose

- JavaScript™ program...contains calls to the server-side software method in accordance with an API and executing the JavaScript™...to call the server-side software.

However, Merrill discloses an analogous process includes script contains calls to the server-side software method in accordance with an API that when executing the script on the client side, calling the server-side software methods ("**browser loads an interpreter in the process space of the browser. To execute the script code, the browser uses the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code... when the script code reference the character control, the browser accesses the animation server...access to the animation server's methods and properties for a particular character to the animation server**" Col 35, line 1-15). Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to include calls to server-side's methods in the JavaScript™. One of the skilled in the art would have been motivated to modify Nelson's process because it allows browser to access animation server's methods and properties for a particular character to the animation server (Col 35, line 1-15).

As per claim 18:

Nelson discloses:

- passing parameters to the server side method when the JavaScript™ is executed on the client side ("**TABLE I" Col 5, line 35-65; the API (Application Program Interface) accepts all the parameters, for instance: vCubeID as integer, vReportID as string**...).

As per claim 19:

Nelson discloses:

- receiving results from the server-side method when the JavaScript™ is executed on the client side ("**upon receiving the script, client device 9 executes the script and builds a presentation model...In addition, client device 9 requests any remaining data that the OLAP servers returned...upon receiving the request, packet engine 21 transmits the remaining data in a series of packets to client device**..." Col 8, line 56-60).

As per claim 20:

Nelson discloses:

- a portion for creating an API on the server side for a ser-side software method ("**API and TABLE I**" Col 5, line 31-65).

- a portion for sending the created JavaScript™ program to the client side ("**transmits the script to web browser**" Col 8, line 54-55).

- a portion for automatically creating a JavaScript™ program ("**generates the client-side script**" Col 8, line 54),

Nelson does not explicitly disclose:

- JavaScript™ contains calls to the server-side software methods in accordance with the API.

However, Merrill discloses an analogous apparatus includes script that contains reference to server-side software method ("**browser loads an interpreter in the process space of the browser. To execute the script code, the browser uses the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code... when the script code reference the character control, the browser accesses the animation server...access to the animation server's methods and properties for a particular character to the animation server**" Col 35, line 1-15).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to include calls to server-side's methods in the JavaScript™. One of the skilled in the art would have been motivated to modify Nelson's process because it allows browser to access animation server's methods and properties for a particular character to the animation server (Col 35, line 1-15).

5. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Merrill et al. (Patent No.: US 6,369,821 B2) as applied to claim 2 above, and further in view of Guthrie et al. (Patent No.: US 6,549,955 B2).

As per claim 3:

Neither Nelson or Merrill disclose:

- wherein executing the JavaScript™ includes creating a Java™ object having the same name as a server-side Java™ bean.

However, Guthrie discloses an analogous process for creating a Java™ object having the same name as a server-side Java™ bean ("**remote proxy class (a category of objects) 23 is generated on client system based on the name, interfaces and methods of subject object 18 which my reside on server system...."reflection is a process that determines what an object can do, how it is defined... reflection mirrors the public view of an object to collect information...the reflection process includes the following: name, list of implemented interface; list of methods; and superclass information.**" Col 5, line 57-59; Col 6, line 39-56)

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson and Merrill's approaches to include the refection process. The modification is obvious because one of the ordinary skilled in the art would have been motivated to include the reflection process to **facilitate the creation of proxies, which resemble objects on the public view, but are very different internally, or privately** (Col 6, line 42-43).

6. Claims 5 and 7-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Mikhail et al. (Pub No.: 2003/0218633 A1).

As per claim 5:

Nelson does not explicitly disclose:

- initially registering the server-side method on the server side, wherein the identifying includes identifying the registered server-side methods.

However, Mikhail discloses an analogous process for registering the server-side method on the server side, wherein the identifying includes identifying the registered server side methods ("**the bean registers itself with the Sybase notification server, specifying a callback method for the desired notification**" paragraph 0042).

Therefore, it would have obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's approach to include registering beans on the server. The modification is obvious because one of the ordinary skilled in the art would have been motivated to register the beans on server for the server to remote the beans interface to the client in form of JavaScript™ object. This allows the programmer to call any of the bean's methods from the JavaScript™.

As per claim 7:

Mikhail discloses:

- wherein a subset of methods are specified, thereby identifying the subset of methods ("**for each Sybase notification to be handled, the application server creates at least one Java™ bean. The bean registers itself...specifying a callback method for the desired notification**" paragraph 0042, which means, there are multiple beans are created and each specifying a callback method. The subset of methods is the subset of callback methods for these beans).

As per claim 8:

Mikhail discloses:

- wherein registration is performed using JSP tags. It is inherent in Mikhail's process because JSP tags are included in Mikhail (paragraph 0034).

7. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Mikhail et al. (Pub No.: 2003/0218633 A1) as applied to claim 5 above, and further in view of Chan et al. (Patent No.: US 6,836,889 B1).

As per claim 6:

Neither Nelson or Mikhail disclose:

- wherein no methods are specified, thereby identifying all methods of a bean.

However, Chan discloses an analogous process includes no methods are specified, thereby identifying all methods of a beans ("**business methods of an enterprise bean that can be accessed by a client program**" Col 2, line 26-27).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include this feature in Nelson and Mikhail's processes. The combination is obvious because one of the skilled in the art would have been motivated to include this feature to allow the tools and programmers to easily recognize and utilize the capabilities of Java™ beans defined according to the Java™ Beans specification (Col 2, line 30-36).

8. Claims 9-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1), in view of Mandal et al. (Patent No.: US 7,043,732 B2).

As per claim 9:

Nelson does not explicitly disclose:

- the API (Application Program Interface) identifying parameters of the method.

However, Mandal discloses an analogous process of using the API (application Program Interface) identifying parameters of the method ("**provides APIs to add or remove a SNDRSet from the SNDRGroup, to lock or unclock a group, to modify group adjustable parameters**..." Col 16, line 42-44).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process to have his API (Application Program Interface) performs identify parameters. One of the skilled in the art would have been motivated to use API (Application Program Interface) for identifying parameters because API (Application Program Interface) allows programmer to add, remove, or modify parameters (Col 16, line 42-44).

As per claim 10:

Nelson does not explicitly disclose:

- allowing specification of which server-side methods are included in the API.

However, Mandal discloses an analogous process of using the API (Application Program Interface) and allowing specification of which server-side methods are included in the API ("**provides APIs to add or remove a SNDRSet (method) from the SNDRGroup (methods)..." Col 16, line 42-43, API (Application Program Interface) identifies which method to add or remove**).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Nelson's process having API (Application Program Interface) to chose which method to include in API. One of the skilled in the art would have been motivated to modify because API (Application Program Interface) allows to remove or to add methods (Col 16, line 42-43).

As per claim 11:

Nelson does not explicitly disclose:

- the method is a method in a Java™ bean.

However, Mandal discloses an analogous process using method in a Java™ bean ("**the SNDR Federated bean 434 (SNDRBean) comprises an implementation 500 that is created by a constructor for** ..." Col 16, line 11-12, SNDRBean is a method that comprises an implementation).

Therefore, it would have been obvious to one having an ordinary skilled in the art at the time the invention was made to modify Nelson's approach to include a method in a Java™ bean. One of the ordinary skilled in the art would have been motivated to use a method of Java™ bean because it is a reusable software component and it can be combined to create an application.

9. Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nelson et al. (Patent No.: US 6,188,400 B1) and Merrill et al. (Patent No.: US 6,369,821 B2) as applied to claim 2 above, and further in view of Mein et al (Patent No.: US 6,457,066 B1).

As per claim 14:

Neither Nelson or Merrill disclose:

- wherein executing the JavaScript™ further includes using SOAP calls to invoke the server-side method.

However, Mein discloses an analogous process that using SOAP calls to invoke the server-side method ("**The protocol, called a Simple Object Access Protocol (SOAP), is an application layer protocol that is layered on top of HTTP and allows Microsoft Component Object Model (COM) Automation objects to be accessed and methods to be invoked over the internet through Web servers protected by firewalls**" Col 3, line 10-15").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches. One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

As per claim 15:

Neither Nelson or Merrill disclose:

- wherein executing the JavaScript™ further includes using SOAP protocol to return results from the server-side method.

However, Mein discloses an analogous process for using SOAP protocol to return results from the server-side method ("**The SOAP stub, which is running on the Web server, unpacks and parses the SOAP request, instantiates the COM Automation object...The SOAP stub also returns any [out] or [in, out], or returns parameters from the COM Automation object instance to the SOAP proxy**..." Col 3, line 54-60).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches. One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

As per claim 16:

Nelson discloses:

- wherein the JavaScript™ includes JavaScript™ instructions for:
    - although, Nelson does not explicitly disclose declaring public methods for a current session. It is inherent in Nelson' process of generating the JavaScript™ in order for the browser to build the presentation model object ("**builds a presentation model**" Col 8, line 57);
    - instantiating JavaScript™ objects on the client-side that correspond to server-side objects ("**upon receiving the script, client device 9 executes the script and builds a presentation model**" Col 8, line 56-57, because the server-side device generated the script and the presentation model object builds based on the script, therefore, it is corresponding to server-side).

Nelson does not explicitly disclose:

- setting up SOAP requests for each server-side method

However, Mein discloses an analogous process for setting up SOAP requests for each server-side method ("**The protocol, called a Simple Object Access Protocol (SOAP), is an application layer protocol that is layered on top of HTTP and allows Microsoft Component Object Model (COM) Automation objects to be accessed and methods to be invoked over the Internet through Web servers protected by firewalls**" Col 3, line 10-15).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to include SOAP in Nelson and Merrill's approaches. One of the skilled in the art would have been motivated to include SOAP in Nelson and Merrill's approaches because using SOAP to access objects and to invoke methods (Col 3, line 10-15).

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

WEI ZHEN
SUPERVISORY PATENT EX

PN
5/14/2007